Chapter 4

Sources

Theodore Gerontakos

An essential part of the AP process is to determine the sources of the components we mix and match. Although it's true that linked-data practices and other technologies have changed our work, there has been an increase in linked-data-ready sources available. The quantity of tools available at the Library of Congress Linked Data Service alone is remarkable, not to mention all the other vocabularies and ontologies available as linked-data resources.

Library of Congress Linked Data Service
https://id.loc.gov/

In this chapter on sources, we'll look at four sources of AP components: ontologies, schemas, vocabulary encoding schemes, and syntax encoding schemes, then conclude with some ideas on how to find them.

Ontologies

Ontologies have emerged as our richest source of AP components. Creating an ontology is difficult, as they often model total knowledge domains. Modeling library data alone is an enormous undertaking that requires many entities and properties. Once complete, however, many ontologies are relatively easy to use for assembling APs. They are essentially lists, with attendant descriptions, of classes and properties. The classes are types for our entities; our entities are instances of a given class. The properties relate the entities to other entities.

The ontologies that concern us here are RDF ontologies. These can be expressed using classes and properties specified for creating descriptions of classes and properties. The core instruments for creating RDF ontologies are found in RDF Schema 1.1 (RDFS).[1] We find the classes Class and Property in RDFS (although Property is technically in the RDF namespace, not RDFS) to classify instances of those classes. We also find properties such as domain, range, subPropertyOf, and subClassOf. RDFS is a small but essential ontology for creating ontologies.

The Web Ontology Language (OWL) is a family of ontology languages that incorporates and extends all of RDFS and vastly increases its expressive power.[2] Using OWL, classes, properties, and complex values of properties can be described using instruments based in formal logic. Most of the ontologies we use to create library data do not dive too deeply into OWL constructs, but, rather, use elements of OWL when they are useful.

An actual RDF ontology is serialized as RDF and usually can be accessed in one of many possible RDF serializations (RDF/XML, Turtle, N-Triples, etc.). In addition, it has become commonplace to make the ontology viewable as a human-friendly HTML page. All modes of access should be available over the World Wide Web. The BIBFRAME ontology is a good example: it can be accessed in many different flavors of RDF using a download link embedded in a web page; it can also be viewed in its totality as an HTML page, with some prefatory material on top, then the class list, followed by the property list, all items in each list hyperlinks to class and property descriptions.[3]

In the previous chapter, we saw how using properties can be complex because, in their original context, they are part of a syndetic structure. Ontologies require the same level of expertise to use accurately. The classes and properties lists are taxonomies whose syndetic structure should be analyzed before they are used. Although ontologies are model-centric conceptualizations that help us *interpret* data, not data-centric rules that validate data—ontologies are not constraints—if we do not create data that agrees with

our source ontologies, the truth value of our instance data will decrease. The difference between the closed world of constraints and the open world of ontologies is subtle and difficult to discern. Sometimes it seems best to understand the concepts in an ontology as if they were constraints.

Currently the primary model for library data is IFLA's *Library Reference Model* (LRM).[4] This is a high-level conceptual model that consolidates three earlier IFLA conceptual models, FRBR, FRAD, and FRSAD. It is intended to be a broad logical structure that more detailed ontologies align with. RDA can be considered an ontology that extends and aligns with LRM.[5] RDA, along with BIBFRAME, has become one of two major ontologies for describing library data.

RDA is a complete ontology for describing thirteen entities crucial to library data, including the famous Group 1 entities inherited from FRBR, namely Work, Expression, Manifestation, and Item. RDA's most striking feature, at first glance, is a large quantity of properties. The resulting instance data created using the RDA ontology, with relatively few classes and many properties, is not layered and nested but quite flat, and so is easily rendered using RDF triples (although RDA does not assume RDA data will be expressed as RDF). In addition to abundant properties, RDA provides detailed instructions on forming values. The large quantity of properties also includes precise relationships between entities, making RDA the only ontology to date that can express the modeling of bibliographic relationships that is one of the great achievements in library science over the past thirty years.

Organizations that intend to use RDA are advised to create APs for its use. The magnitude of properties needs to be narrowed for specific applications, and there are multiple options for forming values and for creating new entities that require rulings (for example, is the item in hand a new work or an expression of an already-existing work?). Exactly what these profiles will look like is undetermined. One of the locations for viewing RDA, the RDA Toolkit, displays some human-readable APs that offer a clue.[6]

BIBFRAME is the second of our two major ontologies. BIBFRAME is not aligned with LRM and lacks the expressive power of RDA when describing relationships. However, BIBFRAME features a rich taxonomy of classes and has de facto become the more widely adopted ontology. Instance data created using BIBFRAME is deeply layered and nested and can be difficult to process and query. It does not model bibliographic relationships in detail, which creates a significant loss of data when converting RDA data to BIBFRAME data. In addition, BIBFRAME uses a different model for describing products of intellectual and artistic endeavors than RDA, eschewing the RDA entity Expression, which has created an incompatibility between the two models. Although some adopters of BIBFRAME aspire to form values in BIBFRAME using RDA rules, RDA data is more accurately rendered using the RDA ontology.

The Linked Data for Production 2 (LD4P2) project participants created APs for use in the Sinopia platform.[7] These APs were authored using the LC BIBFRAME Profiles specification.[8] They can be regarded as state-of-the-art APs. Almost all the APs focus on the implementation of BIBFRAME classes and properties; one notable exception is the University of Washington contribution, which focuses on implementing RDA classes and properties.[9]

Whether using BIBFRAME, RDA, or other ontologies, rarely is a single ontology entirely sufficient to describe our resources. It is common to combine properties and classes from multiple ontologies using an AP, as well as to annotate and refine the uses of those classes and properties. In addition to multiple ontologies, ontology extensions are useful sources for APs; these are mini-ontologies that provide suitable classes and properties for applications not covered by the parent ontology. In one of the original Linked Data for Production initiatives (LD4P), several ontology extensions were written to extend BIBFRAME.[10]

There are many other ontologies that are useful for creating library data. These include the following:

- General
  - Dublin Core
  - Schema.org
  - CIDOC-CRM
  - MODS
  - DPLA
  - Europeana
- Authority data: MADS
- Preservation metadata: PREMIS
- Taxonomies and thesauruses: SKOS
- People
  - FOAF
  - VIVO
- Data sets
  - VoID
  - DCAT

Schemas

Schema is a word used frequently when discussing metadata. Here we are referring to a metadata element set. As usual, however, it is a little more complicated than that. Metadata schemas are something we often referred to in the XML era.[11] Usually *metadata schema* means a complete element set, commonly presented as a document intended for human consumption (an HTML page, a PDF, etc.) backed up by code written

in a schema language (most often XML Schema). The code's main purpose is to validate, or constrain, the data.

Many monumental schemas have been produced, including EAD, METS, MODS, PREMIS, DDI, and many more. These schemas aspire to meet all an application's descriptive needs, and instance metadata can be stored as a machine-actionable XML document or document-like object. Machine actions, or processing, focus on validation, but also could include additional processes such as conversion to other metadata schemas.

Application profiles emerged in the XML era, mostly as an attempt to supplement general metadata schemas such as Dublin Core; however, sometimes even the comprehensive schemas require extensions using elements from other schemas. In most cases, these APs are human-readable documents with a list of properties, the properties of the properties, their source, and sometimes mappings to other schemas. XML tools are powerful and permit metadata professionals to go far beyond human-readable documents; nevertheless, the common practice was to remain practical and meet local needs with appropriate effort. APs created in spreadsheets for immediate local use were common.

These practices persist. It is still common for projects to create only human-readable APs, and we still see instance metadata stored in an XML document, often constrained by an XML schema that accompanies an element set—or even an XML schema based on an AP. Although linked-data practices prefer a totalized machine-actionability and RDF serialization with dereferencing capability in the web, XML-era instance data is still abundant and useful. It can even be used to derive RDF data. As stated, XML tools are powerful, and XML data should persist well into the future.

Over the years many element sets have been rebuilt as ontologies, which is no small feat. The element sets are independent resources with an internal consistency; to rebuild them as stand-alone ontologies, all the properties need to be defined, syndetic structure retained, entities identified—it's a lot of work. In the meantime, many schemas can still be used advantageously, and APs can still be assembled and backed up with local validation code. If nothing else, schemas can be a rich source of ideas for creating APs and other local models. As we've seen above, we continue to describe and borrow elements the same way we have for many years. We've just added a few new practices to improve data integration with the web in the 2020s.

Here is a list of some well-known schemas: EAD, MARCXML, MODS, VRA Core, Dublin Core, OAI-PMH, OAI-ORE, METS, PREMIS, XMP, EXIF, IPTC, MPEG-7, PBCore, DDI, Darwin Core, ONIX, TEI, and MIX.

VES

Librarians are well-acquainted with vocabulary encoding schemes (VESs), also known as value encoding schemes: they are controlled vocabularies—thesauri, taxonomies, classification schemes, subject headings lists, and so on. They are sources for values. The term was adopted at DCMI to refer to complete lists: the complete thesaurus, taxonomy, and so on. This was necessary because individual terms did not have an IRI, and the source of each heading needed to be recorded in our instance data, preferably using an IRI or URL. With the advent of a functioning linked-data infrastructure, many VESs have assigned IRIs to each heading and have described each entry as a discrete resource; reference to the full thesaurus is not as important as it once was in our instance data. Still, VES identification remains an important feature of APs.

Ideally a term from a VES would be represented in our instance data as an IRI. RDA, however, allows terms from VESs to be recorded in our instance data using any of the RDA recording methods (structured, unstructured, identifier, IRI). The VES in this case is seen as a source of values regardless of the node type expected for a value.

RDA offers a range of RDA value vocabularies, or VESs, in the RDA Toolkit and the RDA Registry.[12] The Library of Congress offers a tremendous range of VESs at its Linked Data Service. There are many well-used, widely adopted value vocabularies.

Value vocabularies were considered a sensible place to start creating our linked-data infrastructure. In the 2011 Library Linked Data Incubator Group Final Report, they were singled out as "low-hanging fruit" for conversion to linked data.[13] As a result, there are many value vocabularies that are excellent sources of values, whether they're used for the literals, the identifiers, or the IRIs.

SES

A syntax encoding scheme (SES) is a set of rules for constructing a literal value. As a source for APs, it is less a source of values and more a set of rules for creating a text string.

Many SESs exist. The W3C Date and Time Formats (W3CDTF) is an example.[14] It is a set of rules on how to construct dates. There are rules on how to represent languages, geographic coordinates, access points for information resources, and many more.

DCMI considers an SES a data type.[15] As such, an SES-structured value can be rendered in RDF data as a typed literal, described above in the "Values" section of chapter 3. To facilitate this, Dublin Core Metadata

Terms currently lists twelve SESs and assigns IRIs. Thus for a date structured using W3CDTF, an RDF object node could be entered as follows: "1997-07-07"^^<http://purl.org/dc/terms/W3CDTF>.

RDA calls an SES a "string encoding scheme." The RDA definition is difficult to understand: "A set of string values and an associated set of rules that describe a mapping between that set of strings and a value of an element." [16] Here we maintain the phrase "set of rules" and take comfort that we can think of an SES as a data type.

In our APs, we should specify the source of the set of rules. If convenient, we can simply repeat the rules in the AP. However, in most cases this would unnecessarily overburden our AP with excessive detail when we could simply reference the SES. Again, it is a judgment we need to make to balance simplicity and complexity.

Finding Sources

When we want to create our AP or extend a schema or ontology, how do we find these sources of properties, values, and classes?

In most cases, metadata professionals are just familiar with available resources or find them by querying the web.

An alternative is to consult lists of resources available; librarians are well-known for their lists of resources, and lists of resources for writing APs are available (sometimes embedded in more generalized lists). Here are a few:

- "Semantic Web and Linked Data," UCLA Library, https://guides.library.ucla.edu/semantic-web/semantic_web_vocabularies.
- "Lists of Ontologies," World Wide Web Consortium, last updated December 13, 2013, https://www.w3.org/wiki/Lists_of_ontologies.
- "Metadata for Data Management: A Tutorial," UNC University Libraries, last updated January 25, 2021, https://guides.lib.unc.edu/metadata/standards.
- "Metadata Standard," Wikipedia, last updated April 15, 2021, https://en.wikipedia.org/wiki/Metadata_standard.

The preferred method is to find sources of properties, classes, or values directly on the web, as if we were searching a metadata registry.[17] This is not yet a mainstream practice, but there are some implementations, most notably Linked Open Vocabularies (LOV).[18] Other examples include BioPortal, Biblioportal, and a taxonomy search called TaxoBank.[19]

The bad news is that there is no easy way to find components for an AP. This is another reason AP authoring is performed with the assistance of a specialist.

Notes

1. Dan Brickley and R. V. Guha, eds., "RDF Schema 1.1," section 3.2, W3C Recommendation, World Wide Web Consortium, February 25, 2014, https://www.w3.org/TR/2014/REC-rdf-schema-20140225.
2. OWL Working Group, "OWL: Web Ontology Language (OWL)," World Wide Web Consortium, December 11, 2012, https://www.w3.org/OWL/.
3. "BIBFRAME Ontology," v. 2.0.1, Library of Congress, https://id.loc.gov/ontologies/bibframe.html.
4. Pat Riva, Patrick Le Bœuf, and Maja Žumer, IFLA Library Reference Model: A Conceptual Model for Bibliographic Information (The Hague, Netherlands: International Federation of Library Associations and Institutions, August 2017, last updated December 2017), https://www.ifla.org/publications/node/11412. The LRM is also expressed as RDF at https://www.iflastandards.info/lrm/lrmer.html.
5. RDA Toolkit, https://access.rdatoolkit.org.
6. "Application profiles," RDA Toolkit, https://access.rdatoolkit.org/Guidance/Index?externalId=en-US_ala-591ca278-2807-399b-9530-6b44171e6ccc.
7. "Profiles on GitHub," LD4P2 Linked Data for Production: Pathway to Implementation," last updated March 31, 2019, https://wiki.lyrasis.org/display/LD4P2/Profiles+on+GitHub.
8. "BIBFRAME Profiles: Introduction and Specification," draft for public review, Library of Congress, May 5, 2014, https://www.loc.gov/bibframe/docs/bibframe-profiles.html.
9. Linked Data for Production, "LD4P/sinopia_sample_profiles," GitHub, last updated June 22, 2020, https://github.com/LD4P/sinopia_sample_profiles/tree/master/cohort-profiles/university-of-washington.
10. "LD4P Outputs," Linked Data for Production, last updated August 5, 2018, https://wiki.lyrasis.org/display/LD4P/LD4P+Outputs.
11. Use of XML in libraries flourished ca. 2000–2017—the "XML Era"—but it is still very much alive. The XML tool suite is an outstanding tool set that libraries should continue to use well into the future.
12. "RDA Registry," RDA Steering Committee, in association with ALA Digital Reference, https://www.rdaregistry.info/.
13. W3C Library Linked Data Incubator Group, "4.1.1 Identify Sets of Data as Possible Candidates for Early Exposure as Linked Data," *Library Linked Data Incubator Group Final Report*, World Wide Web Consortium, October 25, 2011, https://www.w3.org/2005/Incubator/lld/XGR-lld-20111025/#Identify_sets_of_data_as_possible_candidates_for_early_exposure_as_Linked_Data.
14. Misha Wolf and Charles Wicksteed, "Date and Time Formats," World Wide Web Consortium, September

15, 1997, https://www.w3.org/TR/NOTE-datetime.
15. "Syntax Encoding Scheme," Glossary, Dublin Core Metadata Initiative, last updated May 6, 2021, https://www.dublincore.org/resources/glossary/#Syntax%20Encoding%20Scheme.
16. "Glossary," RDA Toolkit, https://access.rdatoolkit.org/Glossary.
17. Such as, for example, the Open Metadata Registry at http://metadataregistry.org/.
18. Linked Open Vocabularies (LOV), https://lov.linkeddata.es/dataset/lov/.
19. BioPortal, https://bioportal.bioontology.org/; Bibliophile, http://biblio.ontoportal.org/; TaxoBank Terminology Registry, http://www.taxobank.org/.